Model-Free Reinforcement Learning with Skew-Symmetric Bilinear Utilities

Hugo Gilbert, Bruno Zanuttini, Paolo Viappiani, Paul Weng, Esther Nicart

Model-Free Reinforcement Learning with Skew-Symmetric Bilinear Utilities

Hugo Gilbert[1], **Bruno Zanuttini**[4], **Paolo Viappiani**[1], **Paul Weng**[2,3], **Esther Nicart**[4,5]
[1]Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606, Paris, France
[2]SYSU-CMU Joint Institute of Engineering, Guangzhou, China
[3]SYSU-CMU Shunde International Joint Research Institute, Shunde, China
[4]Caen University, Normandy, France;
[5]Cordon Electronics DS2i, France;
{hugo.gilbert,paolo.viappiani}@lip6.fr, paweng@cmu.edu, {esther.nicart,bruno.zanuttini}@unicaen.fr

Abstract

In reinforcement learning, policies are typically evaluated according to the expectation of cumulated rewards. Researchers in decision theory have argued that more sophisticated decision criteria can better model the preferences of a decision maker. In particular, Skew-Symmetric Bilinear (SSB) utility functions generalize von Neumann and Morgenstern's expected utility (EU) theory to encompass rational decision behaviors that EU cannot accommodate. In this paper, we adopt an SSB utility function to compare policies in the reinforcement learning setting. We provide a model-free SSB reinforcement learning algorithm, *SSB Q-learning*, and prove its convergence towards a policy that is ϵ-optimal according to SSB. The proposed algorithm is an adaptation of fictitious play [Brown, 1951] combined with techniques from stochastic approximation [Borkar, 1997]. We also present some experimental results which evaluate our approach in a variety of settings.

1 INTRODUCTION

In problems of sequential decision-making under uncertainty (often represented as Markov Decision Problems—MDPs [Puterman, 1994]), an agent has to repeatedly choose according to her current state an action whose consequences are uncertain, in order to maximize a certain criterion in the long run. In most cases, the criterion chosen is the expectation of cumulated rewards, but it is not risk sensitive and fails to explain widely observed violations of axioms such as transitivity or von Neumann's independence axiom. One of the aims of decision theory is to provide criteria able to account for such behaviors.

Interestingly, the Skew-Symmetric Bilinear (SSB) utility theory [Fishburn, 1984] defines a family of decision criteria able to represent risk-averse and risk-seeking behaviors, intransitive choices and violations of the independence axiom. In particular it encompasses the expected utility (EU) model [von Neumann and Morgenstern, 1947], which is the most popular risk sensitive criterion in decision theory. In SSB theory a binary functional φ over probability distributions is given, where the sign of $\varphi(\mathbf{p}, \mathbf{q})$ gives the preference between two distributions $\mathbf{p}$ and $\mathbf{q}$. Furthermore, particular choices of φ allow decision criteria that only rely on ordinal pieces of information such as "this trajectory is preferred to this other trajectory" to be used. This property is of interest for MDPs as the optimal policy can be highly sensitive to the reward function, but specifying such a reward function is often difficult even for an expert user.

Such preference-based approaches have received much attention lately [Akrour *et al.*, 2012, Furnkranz *et al.*, 2012, Busa-fekete *et al.*, 2014, Wilson *et al.*, 2012, Wirth and Fürnkranz, 2013, Wirth and Neumann, 2015], and a special case of SSB utility function which optimizes the probability of yielding a preferred outcome has been investigated in various domains [Busa-fekete *et al.*, 2014, Dudík *et al.*, 2015, Rivest and Shen, 2010]. In reinforcement learning (RL), Busa-Fekete *et al.* [2014] provided a meta-heuristic algorithm using evolutionary strategies to compute a "good" policy. Designing a learning scheme for an RL problem using an SSB utility function would therefore enable RL problems to be solved for a large class of criteria including "preference-based" criteria.

The possibility of intransitive preferences in SSB utility theory could be seen as a significant barrier to its use in automated decision making. However, the seminal work of Gilbert *et al.* [2015a] shows that an SSB-optimal strategy always exists as a mixture of policies, and furthermore, that in a finite horizon MDP where the model is known, such an SSB-optimal strategy can be computed using a double oracle approach. Unfortunately, this approach suffers two drawbacks: its time and space requirements might become prohibitive if the optimal mixture of policies is composed of too many policies, and it does not generalize to the case where the model is unknown, and thus cannot be used to derive a model-free RL algorithm.

We remedy this by providing a new algorithm which behaves optimally in an MDP with an SSB utility function, from which we derive a model-free RL algorithm in order to learn an SSB-optimal policy when the model is unknown. While the approach is still based on game-theoretic arguments, the algorithm differs in spirit and resembles an adaptation of a fictitious play algorithm [Brown, 1951] combined with Q-learning [Watkins and Dayan, 1992] using stochastic approximation techniques with two timescales [Borkar, 2008]. Kalathil *et al.* [2014] recently exploited a similar two-timescale technique in MDPs, but in a different context; using the average expected reward criterion with vector rewards, their goal is to learn a policy whose vectorial value approaches a fixed set.

The paper is organized as follows. In Section 2, we give the background elements and introduce our notations. In Section 3, we give a game theoretic view of the resolution of an RL problem with an SSB utility function and present a fictitious play approach to solve it. Lastly, in Section 4, we present some experimental results.

2 SSB MARKOV DECISION PROCESSES

We study in this paper episodic MDPs with finite state and action spaces. An MDP $\mathcal{M}$ is formally defined by a tuple $(\mathcal{S}, \mathcal{F}, \mathcal{A}, \mathcal{P}, s_0)$ where: $\mathcal{S}$ is a finite collection of states; $\mathcal{F} = \{f_1, f_2, \ldots, f_{|\mathcal{F}|}\} \subset \mathcal{S}$ is a finite set of final states; $\mathcal{A} = \{\mathcal{A}_s | s \in \mathcal{S}\}$ is a collection of finite sets of possible actions, one for each state; $\mathcal{P}$ is the transition function where $\mathcal{P}(s'|s,a)$ is the probability that the state at time step $t+1$ is s', given that the state at time step t was s and that the agent performed action a; $s_0 \in \mathcal{S} \setminus \mathcal{F}$ is the initial state in which all episodes start.[1]

Whereas preferences over state-action pairs are typically modeled with numerical rewards, in our framework we assume that the final states summarize the preference information. More precisely, we assume that the decision maker has a preference relation $\succeq$ over possible final states, where $f \succeq f'$ means that ending in final state f is at least as good as ending in f'. Note that *we do not assume $\succeq$ to be total or even transitive*, thus accommodating a wide variety of preference behaviors, including those deviating from normative decision theory. A "standard" MDP (with rewards obtained at each time step) could still be represented in our setting with the notion of an *augmented MDP* [Gilbert *et al.*, 2015a] at the cost of introducing additional states.

We assume (as standard in RL) that $\mathcal{S}$, $\mathcal{A}$, s_0 and $\mathcal{F}$ are known, that $\mathcal{P}$ is unknown and that at each step the agent knows exactly which state she occupies.

We call *episode* a succession of state-action pairs $(s_0, a_0, s_1, \ldots, s_{t-1}, a_{t-1}, s_t)$, starting in s_0 and ending in a final state $s_t \in \mathcal{F}$. When the current episode ends, a new episode starts in state s_0. We further assume that the length of an episode is upper-bounded by a constant $T_{max} \in \mathbb{N}$.

Table 1: Probabilities for Each Gardner Die

	1	2	3	4	5	6
$\mathbf{p}_A$	1/6	0	0	5/6	0	0
$\mathbf{p}_B$	0	0	5/6	0	0	1/6
$\mathbf{p}_C$	0	1/2	0	0	1/2	0

A *policy* π at horizon T indicates which action to perform in each nonfinal state for each time step $t < T$. A policy is *Markovian* if the action depends only on the current state and timestep (otherwise it may depend on all state-action pairs encoutered so far); *deterministic* if it prescribes exactly one action, or *randomized* if it prescribes a probability distribution over actions; *stationary* if the action prescribed does not depend on the timestep. We write Π_s for the set of Markovian stationary deterministic policies.

Importantly, given a set $\Pi = \{\pi^1, \pi^2, \ldots\}$ of policies, we define an enlarged set $\widetilde{\Pi}$ of policies, that denotes the set consisting of *mixtures* of policies, i.e., $\widetilde{\Pi} = \{\tilde{\pi} = (\pi^1|\alpha_1, \pi^2|\alpha_2, \ldots) : \sum_i \alpha_i = 1,\ \alpha_i \geq 0\}$, where $\tilde{\pi}$ is the *mixed policy*[2] that randomly selects policy π^i with probability α_i at the beginning of each episode.

Example 1. *As a running example, we consider a variant of the classical "Gardner dice" two-player game. Each player has three six-sided dice, written A, B, C and biased as shown in Table 1. Players simultaneously choose a die to throw, and whoever rolls the highest number wins.*

It is easy to see that die A rolls higher than B most of the time, so die A should be preferred to B, but B mostly beats C, and C mostly beats A, hence the relation "more likely to win" is cyclic. The optimal policy for this problem is to play dice A, B and C with probabilities $3/13$, $3/13$ and $7/13$, respectively [Gilbert et al.*, 2015a].*

For illustrative purposes, we consider a variant where each player makes sequential decisions: she must first choose whether to throw die A (action a_A) or not (a_{BC}). If she chooses a_{BC}, then she can choose to throw B (action a_B) or C (a_C). Clearly, this does not change the probabilities with which to throw each die in an optimal policy.

This problem can be represented as an MDP where T_{max} is 2, with an initial state s_0 where $\mathcal{A}_{s_0} = \{a_A, a_{BC}\}$; a state s_{BC} (reached by choosing action a_{BC} in s_0); and six final states $\{f_1, \ldots, f_6\}$ representing the numbers rolled. An example transition probability is $\mathcal{P}(f_3|s_{BC}, a_B) = 5/6$, and an example episode is $(s_0, a_{BC}, s_{BC}, a_C, f_5)$. An example (Markovian stationary deterministic) policy is $\pi_B(s_0) = a_{BC}, \pi_B(s_{BC}) = a_B$.

Using similar notation for A and C, the optimal policy is

[1] A probability distribution $\mathcal{P}_0$ over initial states can easily be accommodated by using a dummy state s_0 with one dummy action whose transitions to all the other states are governed by $\mathcal{P}_0$.

[2] Not to be confused with the notion of randomized policies.

the mixed policy $\tilde{\pi}^*$=($\pi_A|3/13$,$\pi_B|3/13$,$\pi_C|7/13$), which dictates that the player draws one of π_A,π_B,π_C at the start of an episode, following it for the whole episode. [3]

Our aim is to find an optimal (defined in the next subsection) policy. Recall that the decision maker has a preference relation $\succeq$ over possible final states; we want to compare policies by considering this preference relation. In other words, we want to lift the preference relation $\succeq$ defined on final states to a preference relation defined on policies.

2.1 COMPARING POLICIES WITH AN SSB UTILITY FUNCTION

We assume throughout the paper that the agent's preferences between probability distributions are described by the SSB model as presented and axiomatized by Fishburn [1984]. In this model, an agent is endowed with a binary functional φ over ordered pairs $(f, f') \in \mathcal{F}^2$ of final states, indicating the intensity with which she prefers f to f', with $f \succeq f' \Leftrightarrow \varphi(f, f') \geq 0$. Functional φ is assumed to be skew-symmetric, i.e., $\varphi(f, f') = -\varphi(f', f)$ and is extended to the space of probability distributions over $\mathcal{F}$ by bilinearity (wrt the mixture operation on distributions). The SSB criterion for comparing $\mathbf{p}$ and $\mathbf{q}$ is then written:

$$\varphi(\mathbf{p}, \mathbf{q}) = \sum_{f, f' \in \mathcal{F}} p(f) q(f') \varphi(f, f') \tag{1}$$

where $p(f)$ (resp. $q(f')$) denotes the probability of reaching the final state f (resp. f') in distribution $\mathbf{p}$ (resp. $\mathbf{q}$). We have $\mathbf{p} \succ \mathbf{q}$ if $\varphi(\mathbf{p}, \mathbf{q}) > 0$ (strict preference), and $\mathbf{p} \sim \mathbf{q}$ if $\varphi(\mathbf{p}, \mathbf{q}) = 0$ (indifference).

Any policy π in an MDP induces a probability distribution $\mathbf{p}^\pi$ over final states (reached after at most T_{max} time steps); $\mathbf{p}^\pi$ is referred to as the *final state distribution* of π. As comparing policies amounts to comparing their induced distributions, we write $\varphi(\pi, \pi')$ for $\varphi(\mathbf{p}^\pi, \mathbf{p}^{\pi'})$ to simplify notation and define the preference relation $\succsim$ over policies:

$$\pi \succsim \pi' \equiv \varphi(\pi, \pi') \geq 0 \tag{2}$$

Example 2 (continued)**.** *The goal of beating the opponent's roll can be expressed as $\varphi(f_m, f_n) = 1$ for $m > n$, -1 for $m < n$, and 0 for $m = n$. The deterministic policies π_A, π_B, π_C amount to rolling the corresponding die, inducing the final state distributions $\mathbf{p}_A, \mathbf{p}_B, \mathbf{p}_C$ of Table 1 over final states. We have (zero entries are irrelevant):*

$$\begin{aligned}
&\varphi(\mathbf{p}_A, \mathbf{p}_B) \\
&= \mathbf{p}_A(f_1)\mathbf{p}_B(f_3)\varphi(f_1, f_3) + \mathbf{p}_A(f_1)\mathbf{p}_B(f_6)\varphi(f_1, f_6) \\
&\quad + \mathbf{p}_A(f_4)\mathbf{p}_B(f_3)\varphi(f_4, f_3) + \mathbf{p}_A(f_4)\mathbf{p}_B(f_6)\varphi(f_4, f_6) \\
&= \frac{1}{6} \cdot \frac{5}{6} \cdot (-1) + \frac{1}{6} \cdot \frac{1}{6} \cdot (-1) + \frac{5}{6} \cdot \frac{5}{6} \cdot 1 + \frac{5}{6} \cdot \frac{1}{6} \cdot (-1) \\
&= 14/36 > 0,
\end{aligned}$$

showing that die A should be preferred to die B.

If we were to define φ by $\varphi(f_m, f_n) = m - n$, the strength of a victory (or defeat) would be taken into account. The policies in this case would be compared wrt the expectation of the roll (as explained below).

The SSB model is very general, as it can take into account choice intransitivity, which is widely observed in practice [Fishburn, 1991]. Moreover, the SSB model can represent different risk attitudes via an adequate choice of φ. For example, it represents risk-averse behavior (in the weak sense) if the certainty equivalent of a distribution $\mathbf{p}$ is less than or equal to its expected value, where the certainty equivalent of a distribution $\mathbf{p}$ is the element f such that $\varphi(\mathbf{p}, f) = 0$ (*i.e.*, $\mathbf{p} \sim f$). Nakamura [1989] shows how to design risk averse and risk seeking SSB utility functions.

The SSB model also encompasses many decision criteria, such as the expectation criterion $\varphi(f, f') = c(f) - c(f')$ (where c denotes a utility/cost function); the probability threshold criterion [Yu *et al.*, 1998] $\varphi(f, f') = 1_{c(f) \geq \tau} - 1_{c(f') \geq \tau}$, which states that $\mathbf{p} \succ \mathbf{q}$ if $\sum_{c(f) \geq \tau} p(f) > \sum_{c(f) \geq \tau} q(f)$ for a threshold $\tau \in \mathbb{R}$; and the dominance relation $\varphi(f, f') = 1$ (resp. $0, -1$) if $f \succ f'$ (resp. $f \sim f'$, $f \prec f'$), which states that $\mathbf{p} \succ \mathbf{q}$ if $\sum_{f \succ f'} p(f)q(f') > \sum_{f' \succ f} p(f)q(f')$ (as in Example 2). In other words $\mathbf{p}$ is preferred to $\mathbf{q}$ if a final state generated according to $\mathbf{p}$ is more likely to be preferred to a final state generated according to $\mathbf{q}$ than the converse. This is called *probabilistic dominance* in the following.

Probabilistic Dominance (PD) is interesting as it only relies on ordinal pieces of information. Its axiomatic characterization was given by Blavatskyy [2006] and it has been explored lately in various domains such as RL [Busa-fekete *et al.*, 2014], voting systems [Rivest and Shen, 2010] and dueling bandits [Dudík *et al.*, 2015]. In the latter work, the authors adopt the name of *von Neumann solution* for the PD optimal solution. Indeed, as will be discussed in the next section, finding an SSB-optimal policy (and in particular, finding an optimal policy according to PD) is equivalent to finding a Nash equilibrium in a finite zero-sum two-player game. Thus the existence of a von Neumann solution is implied by von Neumann's minimax theorem.

In standard MDPs, the optimal policy can be highly sensitive to the reward function used, and yet designing a numerical reward function is often cognitively difficult, even

[3] Note the difference with the *randomized* (stationary) policy π which draws a_A with probability $3/13$ in s_0, and, *independently of this*, draws a_B (resp. a_C) with probability $3/10$ (resp. $7/10$) in s_{BC}. The expectation of reaching each final state is nonetheless the same in π and $\tilde{\pi}^*$ (hence π is also optimal).

for an expert user. This issue is tackled in preference-based approaches [Akrour *et al.*, 2012, Weng and Zanuttini., 2013, Gilbert *et al.*, 2015b, Weng *et al.*, 2013, Busa-fekete *et al.*, 2014, Furnkranz *et al.*, 2012, Wilson *et al.*, 2012, Wirth and Fürnkranz, 2013, Wirth and Neumann, 2015] by only considering ordinal pieces of information, such as feedbacks of the type "this trajectory is preferable to that one". We can distinguish two types of approach. The first [Wirth and Fürnkranz, 2013, Wirth and Neumann, 2015, Weng *et al.*, 2013] aims to recover a numerical reward function that explains most of the expressed preferences of the user and can be used with classic criteria. The second [Busa-fekete *et al.*, 2014, Furnkranz *et al.*, 2012] deals with purely ordinal criteria, *e.g.*, Busa-Fekete *et al.* [2014] find a good policy (wrt to PD) using a meta-heuristic algorithm based on evolutionary strategies in finite horizon continuous MDPs.

Example 3. *Consider a car racing against the clock. The driver aims to complete the race in the shortest time possible, but must find the right compromise between speed and the risk of being eliminated; driving too fast can cause the car to run off the track and be eliminated from the competition; the relation between the car's speed and going off track is stochastic (the faster, the more likely). This is easily modeled as an MDP where policies induce a distribution over the race's possible outcomes. The preference $\succ$ over final states is determined by two conditions: (i) race completion is always strictly preferred to an elimination, and (ii) a trajectory completing the race in time t_1 is strictly preferred to completion in time $t_2 > t_1$.*

In this model, adopting SSB with probabilistic dominance finds the "best" policy for a driver who wants to maximize her chance of winning a race (against other drivers facing the same MDP). In contrast, traditional approaches would solve this problem by setting a (large) negative reward r_{elim} for an elimination and a small negative reward for each time step before completing the race, and then maximize expectation of rewards using, for example, Q-learning or any other classic algorithm. Different values of r_{elim} would result in very different policies, and while it might be possible to find a good compromise value allowing for a competitive behavior, manually tuning the reward function would be difficult in more complex scenarios.

Consequently, designing and implementing an algorithm for solving MDPs in an RL setting with an SSB utility function also gives us a tool to compute optimal policies for a large class of criteria, including "preference based" ones. A first step towards the design of this algorithm is to give a game-theoretic view of the problem.

2.2 A GAME ON POLICIES

When an MDP is fixed, Equations 1 and 2 induce a zero-sum two-player symmetric game where the set of strategies coincides with the set of possible policies. The players $i \in \{1, 2\}$ simultaneously choose a strategy $\tilde{\pi}_i$ (pure or mixed). The resulting payoff is then given by $\varphi(\tilde{\pi}_1, \tilde{\pi}_2)$. As emphasized by Gilbert *et al.* [2015a], an SSB optimal policy can be found by computing a Nash equilibrium of this game. Indeed, Nash equilibria $(\tilde{\pi}^*, \tilde{\pi}^*)$ are characterized by $\forall \tilde{\pi}, \varphi(\tilde{\pi}^*, \tilde{\pi}) \geq 0$.

Gilbert *et al.* [2015a] also showed that in this game, a best response to a strategy $\tilde{\pi}$ is given by a policy maximizing the expectation of cumulated rewards with reward function:

$$\mathcal{R}_{\mathbf{p}^{\tilde{\pi}}}(s) = \begin{cases} \mathbf{1}_i^\top \Phi \mathbf{p}^{\tilde{\pi}} & \text{if } s = f_i \in \mathcal{F} \\ 0 & \text{otherwise} \end{cases}$$

where $\mathbf{1}_i$ is the i^{th} canonical vector, $^\top$ is the transpose operator and Φ is the SSB matrix (i.e., $\Phi_{i,j} = \varphi(f_i, f_j)$). Put another way, the reward obtained when arriving in the i^{th} final state is given by the i^{th} element of the vector $\Phi \mathbf{p}^\pi$ (the reward is 0 for nonfinal states). Since this defines a standard MDP, there is always a stationary, Markovian and deterministic optimal policy (i.e., the best response to $\tilde{\pi}$). Thus we can restrict ourselves to the finite game with Π_s as the set of pure strategies. To summarize, a Nash equilibrium of this game will give an SSB-optimal policy, in the form of a mixed policy $\tilde{\pi}^* \in \tilde{\Pi}_s$.

Example 4 (continued)**.** *The game induced by our running example has pure strategies π_A, π_B, π_C for both players, with payoff, e.g., $\varphi(\pi_A, \pi_B) = 14/36$ (Example 2). Consider the mixed policy $\tilde{\pi}_{AAB} = (\pi_A|2/3, \pi_B|1/3)$; Table 1 gives its final state distribution $\mathbf{p}^{\tilde{\pi}_{AAB}} = (2/18, 0, 5/18, 10/18, 0, 1/18)$. The value of response π_A to $\tilde{\pi}_{AAB}$ is given by*

$$\mathbf{p}^{\pi_A} \cdot \Phi \cdot \mathbf{p}^{\tilde{\pi}_{AAB}}$$
$$= (\frac{1}{6}, 0, 0, \frac{5}{6}, 0, 0)^\top (-\frac{16}{18}, -\frac{14}{18}, -\frac{9}{18}, \frac{6}{18}, \frac{16}{18}, \frac{17}{18}) = \frac{7}{54}$$

Similarly, the values of π_B, π_C are $-7/27$ and $1/18$ respectively, so a best response to $\tilde{\pi}_{AAB}$ is π_A. The second component of vector $\Phi \cdot \mathbf{p}^{\tilde{\pi}_{AAB}}$ can be computed as follows. We know that rolling 2 is worse than $16/18$ and better than $2/18$ of the outcomes of $\mathbf{p}^{\tilde{\pi}_{AAB}}$, and $\varphi(f_i, f_j)$ is always 1 for $i > j$. Therefore this component is $-1 \cdot 16/18 + 1 \cdot 2/18 = -14/18$.

In a finite zero-sum symmetric game, it is well-known that there exists a symmetric Nash Equilibrium (NE). We aim to compute this NE on policies characterized by payoff function φ (*solving the game*).

3 SOLVING THE GAME

Games in strategic forms can be solved by linear programming [Chvátal, 1983], unfortunately, here, the game is too large to be solved directly (we remind the reader that the number of pure strategies of the game is equal to the number of deterministic stationary policies of the MDP, which

is exponential in the number of states).[4] If the model was known, one could rely on the double oracle algorithm of Gilbert *et al.* [2015a]. Unfortunately, that approach suffers some drawbacks. Firstly, if the optimal mixture of policies $\tilde{\pi}^*$ is composed of too many policies, its time and space requirements might become prohibitive, as the double oracle algorithm would have to compute and store all polices that are in the support of $\tilde{\pi}^*$. Secondly, this approach does not generalize to the case where the model is unknown. Thus it can not be used to derive a model-free RL algorithm. We therefore turn to a different approach, based on fictitious play and on a double timescale technique.

3.1 LEARNING SETTING

In RL, one typically expects convergence to playing an optimal policy at each step. Since in our case the optimal policy is mixed, this cannot be evaluated at each time step independently. Rather, after any number n of episodes, we consider the *final state distribution*, defined as $(f_1|\alpha_1, \ldots, f_{|\mathcal{F}|}|\alpha_{|\mathcal{F}|})$, where for all i, α_i is the fraction of episodes so far in which the final state was f_i.

We define the *loss* $L(\mathbf{p})$ of a distribution over final states to be the value of its best response against it:

$$L(\mathbf{p}) \stackrel{def}{=} \varphi(\mathbf{p}^{BR}, \mathbf{p})$$

where $\mathbf{p}^{BR}$ is the vector of final state frequencies of a policy which maximizes the expectation of cumulated rewards with respect to reward funtion $\mathcal{R}_{\mathbf{p}}$ (hence a best response to $\mathbf{p}$). Since $L(\mathbf{p}^{\tilde{\pi}^*}) = 0$ characterizes SSB-optimal policies $\tilde{\pi}^*$, it is natural to measure the quality of learning by the decrease in loss of the final state distribution so far $\mathbf{p}_n$, as n increases. Convergence to an SSB-optimal policy then amounts to $L(\mathbf{p}_n) \to 0$ with $n \to \infty$.

Rephrasing, we expect that, considering the final states reached from the beginning, in retrospect their frequencies are approximately equivalent to those we would have obtained had we played a mixed optimal policy from the start (and more and more exactly as n increases). This corresponds to the standard "on-line" setting of RL, in which success is measured from the start.

3.2 FICTITIOUS PLAY

Fictitious Play is an algorithm that only needs a best response procedure to solve a game. The algorithm maintains for each player her mixed policy so far $\tilde{\pi}_n$, defined as $(\pi^1|\alpha_1, \ldots, \pi^k|\alpha_k)$, where for all i, α_i is the fraction of episodes so far in which the stationary, deterministic policy π^i has been played. At each time step, each player considers that $\tilde{\pi}_n$ perfectly represents the mixed strategy that is used by the adversary and plays a best response to it. The algorithm converges to a Nash equilibrium of the game (in the sense that $L(\mathbf{p}_n)$ converges to 0) when the game is a finite zero-sum game. Fictitious play is represented in Algorithm 1 for a symmetric two-player zero-sum game. As the game is symmetric, we only need to consider the mixed policy so far, $\tilde{\pi}_n$, of a player playing against herself.

Algorithm 1: Fictitious Play

Data: Game $\mathcal{G}$, arbitrary pure strategy π_0

1 **while** *True* **do**

2 Play π_n

3 # update current mixed policy

4 $\tilde{\pi}_{n+1} = (\pi^1|\alpha_1, \ldots, \pi^k|\alpha_k)$ with

5 $\begin{cases} \alpha_i = n \cdot \alpha_i/(n+1) + 1/(n+1) & \text{for } \pi^i = \pi_n \\ \alpha_i = n \cdot \alpha_i/(n+1) & \text{for } \pi^i \neq \pi_n \end{cases}$

6 $\pi_{n+1} = BestResponseTo(\tilde{\pi}_{n+1})$

Example 5 (continued)**.** *Assume the agent chooses initial strategy $\pi_0 = \pi_B$, then after one episode/game we have $\tilde{\pi}_1 = (\pi_B|1)$. Now π_A is a best response to $\tilde{\pi}_1$ (Example 2), hence $\pi_1 = \pi_A$. So the agent plays π_A during the second episode, and we get $\tilde{\pi}_2 = (\pi_A|1/2, \pi_B|1/2)$. Now it is easy to see that π_A is a best response to $\tilde{\pi}_2$, so the agent again plays π_A and gets $\tilde{\pi}_3 = (\pi_A|2/3, \pi_B|1/3)$. From Example 4 we get that π_A is a best response to $\tilde{\pi}_3$, and that the loss of $\tilde{\pi}_3$ is $L(\mathbf{p}^{\tilde{\pi}_3}) = \varphi(\pi_A, \tilde{\pi}_3) = 7/54$.*

3.3 SSB Q-LEARNING

This subsection makes a first step towards the adaptation of fictitious play to solve the game induced by an MDP and an SSB utility function. Recall from Section 2.2 that the best responses to the mixed strategy so far, $\tilde{\pi}_n$, are exactly the optimal policies in the (standard) MDP with reward function $\mathcal{R}_{\mathbf{p}^{\tilde{\pi}_n}}$. In other words, best responses can be computed as a function of $\mathbf{p}^{\tilde{\pi}_n}$ only. Accordingly, instead of recording $\tilde{\pi}_n$ as such, which involves an exponential number k of pure policies π^i in the worst case, as in Algorithm 1, it is enough to record the vector $\mathbf{p}_n = \mathbf{p}^{\tilde{\pi}_n}$. Then we can rewrite Lines 4–6 of Algorithm 1 as:

$$\begin{aligned} \mathbf{p}_{n+1} &= n \cdot \mathbf{p}_n/(n+1) + \mathbf{p}^{\pi_n}/(n+1) \\ \pi_{n+1} &= BestResponseTo(\mathbf{p}_{n+1}) \end{aligned}$$

However, in practice, when policy π_n is played, one observes only one drawing from $\mathbf{p}^{\pi_n}$, and not the distribution itself. Hence our first adaptation of fictitious play is as given in Algorithm 2. Note that we do not know the model of the MDP, so we use $BestResponseTo$ as an oracle. We find a better solution later in this section.

Example 6 (continued)**.** *Let $\pi_0 = \pi_B$ again. Hence the agent plays π_B during one complete episode. If the die rolls 3, we get $\mathbf{p}_1 = (f_3|1)$. Then the best response is computed as one to a strategy which always yields f_3, and we get $\pi_1 = \pi_A$. The agent therefore plays π_A during one episode,*

[4] Our running example does not illustrate this combinatorial explosion, but it clearly arises, e.g., in the "intransitive grid" and the "race against the clock" (see Section 4).

Algorithm 2: Approximate Fictitious Play

Data: MDP $\mathcal{M}$, SSB function φ, arbitrary policy $\pi_0 \in \Pi_s$

while *True* **do**
- Play π_n for one episode
- $f_i =$ final state reached
- $\mathbf{p}_{n+1} = n \cdot \mathbf{p}_n/(n+1) + \mathbf{1}_i/(n+1)$
- $\pi_{n+1} = BestResponseTo(\mathbf{p}_{n+1})$

and if the die rolls 4*, we get* $\mathbf{p}_2 = (f_3|1/2, f_4|1/2)$*, to which a best response is* $\pi_2 = \pi_A$*. If the die then rolls* 1*, we get* $\mathbf{p}_3 = (f_1|1/3, f_3|1/3, f_4|1/3)$*. In this case the best response is* π_A*, just as in Example 4, but note that it has an estimated value* $\mathbf{p}_A \cdot \Phi \cdot \mathbf{p}_3 = 5/6 \cdot 2/3 - 1/6 \cdot 2/3 = 4/9$*, instead of* $\varphi(\pi_A, \tilde{\pi}_{AAB}) = 7/54$ *if* $\mathbf{p}^{\tilde{\pi}_{AAB}}$ *were directly observed.*

The following theorem proves that observing only realizations of $\mathbf{p}^{\pi_n}$ does not prevent the convergence of $(\mathbf{p}_n)_{n\in\mathbb{N}}$ to the distribution of an SSB-optimal policy.

Theorem 1. *In Algorithm 2,* $L(\mathbf{p}_n)$ *tends to* 0 *as* $n \to \infty$*.*

Proof. Assume an optimal policy π_n with respect to $\mathcal{R}_{\mathbf{p}_n}$ is played during the $(n+1)$-th episode. At the end of the episode, a final state f_i is reached and $\mathbf{p}_{n+1}$ is defined by:

$$\mathbf{p}_{n+1} = \frac{n \cdot \mathbf{p}_n}{n+1} + \frac{\mathbf{1}_i}{n+1} = \mathbf{p}_n + \frac{1}{n+1}(\mathbf{1}_i - \mathbf{p}_n)$$

We rewrite this equation in the following way :

$$\mathbf{p}_{n+1} = \mathbf{p}_n + \frac{1}{n+1}(\mathbf{p}^{\pi_n} - \mathbf{p}_n + \boldsymbol{M}_{n+1}) \qquad (3)$$

where $\boldsymbol{M}_{n+1} = \mathbf{1}_i - \mathbf{p}^{\pi_n}$ is a square integrable martingale difference sequence. Equation 3 is a standard single timescale process with continuous differential inclusion:

$$\dot{\mathbf{p}}(t) \in \{\mathbf{p}^{\pi} - \mathbf{p}(t) : \pi \in \Pi(\mathbf{p}(t))\} \qquad (4)$$

where $\Pi(\mathbf{p})$ denotes the set of optimal policies with respect to reward $\mathcal{R}_{\mathbf{p}}$. A similar differential inclusion can be obtained for the standard fictitious play. However, here, as $\mathbf{p}_n$ is a distribution over final states (not over strategies) and as only a realization of $\mathbf{p}^{\pi}$ is observed, we need to invoke a stochastic approximation argument.

Indeed, the best response correspondence is upper-semicontinuous, with closed and convex values. Hence the existence of at least one solution $\mathbf{p}(t)$ through each initial value $\mathbf{p}(0)$, which is Lipschitz continuous and defined for all positive times, is guaranteed [Hofbauer, 1995]. Let $\mathbf{p}(t)$ be a solution of inclusion 4 and $\boldsymbol{\zeta}(t) = \mathbf{p}(t) + \dot{\mathbf{p}}(t) = \mathbf{p}^{\pi}$ for a best response $\pi \in \Pi(\mathbf{p}(t))$. By definition of L, we have $L(\mathbf{p}(t)) = \varphi(\boldsymbol{\zeta}(t), \mathbf{p}(t))$, and by the envelope theorem:

$$\frac{d}{dt}L(\mathbf{p}(t)) = \frac{\partial\varphi(\boldsymbol{\zeta}(t), \mathbf{p}(t))}{\partial\boldsymbol{\zeta}}\dot{\boldsymbol{\zeta}}(t) + \frac{\partial\varphi(\boldsymbol{\zeta}(t), \mathbf{p}(t))}{\partial\mathbf{p}}\dot{\mathbf{p}}(t)$$

As $\boldsymbol{\zeta}(t)$ maximizes $\varphi(., \mathbf{p}(t))$, the first term is null [Mas-Colell *et al.*, 1995, pp. 964–965], and by linearity:

$$\begin{aligned}\frac{d}{dt}L(\mathbf{p}(t)) &= \varphi(\boldsymbol{\zeta}(t), \dot{\mathbf{p}}(t)) \\ &= \varphi(\mathbf{p}(t), \dot{\mathbf{p}}(t)) + \underbrace{\varphi(\dot{\mathbf{p}}(t), \dot{\mathbf{p}}(t))}_{=0} \\ &= \varphi(\mathbf{p}(t), \dot{\mathbf{p}}(t)) + \underbrace{\varphi(\mathbf{p}(t), \mathbf{p}(t))}_{=0} \\ &= \varphi(\mathbf{p}(t), \boldsymbol{\zeta}(t)) = -\varphi(\boldsymbol{\zeta}(t), \mathbf{p}(t))\end{aligned}$$

as Φ is skew-symmetric (hence $\varphi(\mathbf{p}(t), \mathbf{p}(t)) = \varphi(\dot{\mathbf{p}}(t), \dot{\mathbf{p}}(t)) = 0$, and $\varphi(\mathbf{p}(t), \boldsymbol{\zeta}(t)) = -\varphi(\boldsymbol{\zeta}(t), \mathbf{p}(t))$). Thus, $\frac{d}{dt}L(\mathbf{p}(t)) = -L(\mathbf{p}(t))$, $L(\mathbf{p}(t)) = L(\mathbf{p}(0))e^{-t}$, and hence $L(\mathbf{p}(t))$ tends to 0 with $t \to \infty$. Thus, the set of final state distributions of the optimal strategies is globally attracting for the best response dynamic, which implies the convergence of the discrete stochastic approximation (3) [Benaïm *et al.*, 2006, Properties 1, 2]. □

If the model was known, one could use Algorithm 2. At each iteration of the **while** loop, the corresponding MDP would be solved using dynamic or linear programming to find the best response policy π_{n+1}. Unfortunately, as the model is unknown, this policy can not be computed directly and has to be learned. The "$\pi_{n+1} = BestResponseTo(\mathbf{p}_{n+1})$" line should therefore be replaced by, say, a Q-learning phase which converges asymptotically to the desired policy.

Recall that an agent using Q-learning (in a standard MDP) maintains an estimate of Q-values $Q(s, a)$ using:

$$\begin{aligned}&Q_{n+1}(s_n, a_n) = Q_n(s_n, a_n) \\ &+ \alpha_n(s_n, a_n)(r_{n+1} + \max_b\{Q_n(s_{n+1}, b)\} - Q_n(s_n, a_n))\end{aligned}$$

after taking action a_n in state s_n, ending up in state s_{n+1}, and observing the (numerical) reward r_{n+1}, and where $\alpha_n(s_n, a_n)$ is the value of the learning rate for (s_n, a_n) at timestep n. Note that from now on, for simplicity, we use n to denote the number of time steps (while previously it denoted the number of episodes). Exploitation is realized by choosing $a_n = \arg\max_a Q_n(s_n, a)$ at state s_n. Q-learning is known to converge to an optimal policy in the limit, provided the reward function is stationary [Watkins and Dayan, 1992].

Hence the idea is to adapt Algorithm 2 to run Q-learning while keeping its target reward function fixed to $\mathcal{R}_{\mathbf{p}_n}$ as when it started, and when it has converged (to an approximate best response to $\mathbf{p}_n$), to update $\mathbf{p}_n$ using what has been observed in this phase and start a new Q-learning phase. Unfortunately, the number of learning episodes required to learn an optimal or ϵ-optimal policy is unknown. Therefore, we want to avoid alternating the Q-learning phase and the reward update phase by using a technique from Borkar [1997] which interleaves both phases successfully using a two-timescale approach. In this approach, the

following iterative equations are used concurrently:

$$\mathbf{p}_{n+1} = \mathbf{p}_n + \beta_n(\mathbf{1}_i - \mathbf{p}_n) \text{ when } f_i \in \mathcal{F} \text{ is reached} \quad (5)$$

$$Q_{n+1}(s_n, a_n) = Q_n(s_n, a_n) + \alpha_n(s_n, a_n)\Delta_{n+1} \quad (6)$$

$$\Delta_{n+1} = \mathcal{R}_{\mathbf{p}_n}(s_{n+1}) + \max_b\{Q_n(s_{n+1}, b)\} - Q_n(s_n, a_n)$$

$$\forall s, a, \left\{ \begin{array}{l} \sum_{n=0}^{\infty} \alpha_n(s,a) = \infty \text{ and } \sum_{n=0}^{\infty} \beta_n = \infty \\ \beta_n/\alpha_n(s,a) \to 0 \\ \sum_{n=0}^{\infty} \alpha_n(s,a)^2 + \beta_n^2 < \infty \end{array} \right\} \quad (7)$$

The intuition is that Q_n evolves more quickly than $\mathbf{p}_n$, giving time for Q_n, and hence the best response, to adapt to changes in the target reward $\mathcal{R}_{\mathbf{p}_n}$.

Write $\eta_{\mathbf{p}}$ (resp. $\eta_{s,a}$) for the number of times vector $\mathbf{p}_n$ (resp. state action-pair (s,a)) has been updated (resp. experienced). In our case, β_n is $1/(\eta_{\mathbf{p}}+1)$ since Equation 5 tracks the final state distribution so far, so we can set α_n to $1/(\eta_{s,a}+1)^{2/3}$ for instance. Note that this example works because $\mathbf{p}_n$ is updated after at most T_{max} steps, bounding the number of updates of α_n (for any (s,a)) between two consecutive updates of β_n. With these conditions, the two recursive equations on $\mathbf{p}_n$ and Q_n form a two-timescale stochastic approximation iteration, where $\mathbf{p}_n$ is on a slower timescale than Q_n. The following example gives the intuition behind the two time-scale technique; the convergence proof under Condition (7) is given in Theorem 2.

Example 7. *Suppose that at some point the Q-values of a_A and a_{BC} in s_0 are 0.1 and 0.2, respectively, and that those of a_B,a_C in s_{BC} are 0.3,-0.7. Assume moreover $\mathbf{p}_n$=$(1/5, 0, 1/5, 1/5, 1/5, 1/5)$, $n = 80$ (with 20 episodes of length 1 and 30 of length 2, hence $\eta_{\mathbf{p}} \simeq 0.02$) $\eta_{s_0,a_{BC}} = 30$ ($\alpha_n(s_0, a_{BC}) \simeq 0.1$) and $\eta_{s_{BC},a_B} = 20$ ($\alpha_n(s_{BC}, a_B) \simeq 0.13$).*

Hence the agent will choose a_{BC} in s_0, reach s_{BC} and update $Q(s_0, a_{BC})$ to $\simeq 0.2 + 0.1 \cdot (0 + 0.3 - 0.2) = 0.21$. In s_{BC} she will choose a_B and, if she rolls 3, update $Q(s_{BC}, a_B)$ to $\simeq 0.3 + 0.13 \cdot (\mathbf{1}_3 \cdot \Phi \cdot \mathbf{p}_n + 0 - 0.3) = 0.3 + 0.13 \cdot (-2/5 - 0.3) \simeq 0.2$, and $\mathbf{p}_n$ to $(10/51, 0, 11/51, 10/51, 10/51, 10/51)$.

Note that Q-values have evolved significantly more than $\mathbf{p}_n$ (and hence than the target numerical reward $\mathcal{R}_{\mathbf{p}_n}$).

3.4 HANDLING EXPLORATION

In order for Q-learning to converge, one needs to ensure that all state-action pairs are performed infinitely often. Usually, this exploration is guaranteed through some randomization, using an ϵ-greedy strategy, for instance. We present in this subsection an exploration strategy called *Episodic-ϵ-Greedy (EG for short)* that guarantees that we converge to an ϵ-optimal SSB strategy using the algorithm described by Equations 5 and 6. During learning, an episode will be generated using either the current best policy (defined by the Q-values), with probability $(1-\epsilon)$, or the uniformly random policy, which we denote by π_U, with probability ϵ. If an episode is generated using π_U (i.e., the agent is exploring), then the update of Equation 5 is not performed at the end of the episode. This guarantees that the convergence of $\mathbf{p}_n$ is not biased by the exploration strategy. The final proposed algorithm is presented in Algorithm 3.

Algorithm 3: SSB Q-learning

Data: MDP $\mathcal{M}$, SSB function φ

while *True* **do**
- Choose a_n using the EG exploration strategy
- Play a_n, observe s_{n+1}, and let $r_{n+1} = \mathcal{R}_{\mathbf{p}_n}(s_{n+1})$
- $Q_{n+1}(s_n, a_n) = Q_n(s_n, a_n) + \alpha_n(s_n, a_n)(r_{n+1} + \max_b\{Q_n(s_{n+1}, b)\} - Q_n(s_n, a_n))$
- **if** $s_{n+1} = f_i \in \mathcal{F}$ *and exploration is off* **then**
  - $\mathbf{p}_{n+1} = \mathbf{p}_n + \frac{1}{\eta_{\mathbf{p}}+1}(\mathbf{1}_i - \mathbf{p}_n)$

We are now ready to prove the convergence of SSB Q-learning to an ϵ-optimal policy through two theorems.

Theorem 2. *Under Conditions (7) with $\beta_n = 1/(\eta_{\mathbf{p}}+1)$, in Algorithm 3, $L(\mathbf{p}_n)$ tends to 0 almost surely as $n \to \infty$.*

Proof. The idea is that $\mathbf{p}_n$ can be viewed as quasi-static compared to Q_n. Indeed, let π_n be the greedy policy given by Q_n. We can rewrite the equations as:

$$\mathbf{p}_{n+1} = \mathbf{p}_n + \alpha_n(\epsilon_n + \mathbf{M}'_n) \quad (8)$$

$$Q_{n+1}(s_n, a_n) = Q_n(s_n, a_n) + \alpha_n(s_n, a_n)(T(Q_n)(s_n, a_n) - Q_n(s_n, a_n) + \mathbf{M}''_{n+1})$$

where $\epsilon_n = \frac{\beta_n}{\alpha_n}(\mathbf{p}^{\pi_n} - \mathbf{p}_n)$ and $\mathbf{M}'_n = \frac{\beta_n}{\alpha_n}(\mathbf{1}_i - \mathbf{p}^{\pi_n})$

$$T(Q_n)(s,a) = \sum_{s'} \mathcal{P}(s'|s,a)(\mathcal{R}_{\mathbf{p}_n}(s') + \max_b\{Q_n(s',b)\})$$

$$\mathbf{M}''_{n+1} = \mathcal{R}_{\mathbf{p}_n}(s_{n+1}) + \max_b\{Q_n(s_{n+1}, b)\} - T(Q_n)(s_n, a_n)$$

Clearly $\epsilon_n \to 0$ almost surely ($||\mathbf{p}^{\pi_n}||$ and $||\mathbf{p}_n||$ are bounded). Then $(\mathbf{p}_n, Q_n)$ will converge to the internally chain transitive invariant set of the ODE [Borkar, 2008]:

$$\dot{\mathbf{p}}(t) = 0 \qquad \dot{Q}(t) = T(Q(t)) - Q(t)$$

Let $Q^*(\mathbf{p}_n)$ denote the optimal Q-value function for reward function $\mathcal{R}_{\mathbf{p}_n}$. Therefore $Q_n - Q^*(\mathbf{p}_n) \to 0$ almost surely, which entails that $(\mathbf{p}_n, \pi_n)$ converges to the set $(\mathbf{p}, \pi^*(\mathbf{p}))$ with $\pi^*(\mathbf{p})$ a best response to $\mathbf{p}$. We then rewrite (5) to:

$$\mathbf{p}_{n+1} = \mathbf{p}_n + \beta_n(\mathbf{p}^{\pi^*(\mathbf{p}_n)} - \mathbf{p}_n + (\mathbf{p}^{\pi_n} - \mathbf{p}^{\pi^*(\mathbf{p}_n)}) + (\mathbf{1}_i - \mathbf{p}^{\pi_n}))$$

As $\mathbf{p}^{\pi_n} - \mathbf{p}^{\pi^*(\mathbf{p}_n)} \to 0$ almost surely, the asymptotic behavior is the same as in Theorem 1. Thus the loss of $\mathbf{p}_n$ converges to 0 with $n \to \infty$. □

The result of Theorem 2 uses the fact that with the EG exploration strategy, exploration has no impact on $\mathbf{p}_n$. The drawback of this strategy is that $\mathbf{p}_n$ does not truly represent the frequencies with which each final state has been obtained. If we let $\mathbf{p}_n^{real}$ represent the vector of true frequencies with which each final state has been obtained, then following theorem proves that $\mathbf{p}_n^{real}$ converges to the final state frequencies of an ϵ- optimal SSB-policy.

Theorem 3. *Under Conditions (7) with $\beta_n = 1/(\eta_{\mathbf{p}} + 1)$, when Algorithm 3 is run, $\mathbf{p}_n^{real}$ converges to the final state distribution of an ϵ-optimal SSB-policy almost surely.*

Proof. Let ϵ' denote the parameter of EG exploration and let $\mathbf{p}^{real} = \lim_{n\to\infty} \mathbf{p}_n^{real}$. Then asymptotically we have

$$\mathbf{p}^{real} = (1-\epsilon')\mathbf{p}^* + \epsilon'\mathbf{p}^{\pi_U}$$

where $\mathbf{p}^*$ is the final state distribution of an optimal SSB-policy. Thus for any policy π:

$$\begin{aligned}\varphi(\mathbf{p}^{real}, \mathbf{p}^\pi) &= (1-\epsilon')\underbrace{\varphi(\mathbf{p}^*, \mathbf{p}^\pi)}_{\geq 0} + \epsilon'\underbrace{\varphi(\mathbf{p}^{\pi_U}, \mathbf{p}^\pi)}_{\geq -\varphi_{\max}} \\ &\geq -\epsilon'\varphi_{\max}\end{aligned}$$

with $\varphi_{\max} = \max_{f,f'} \varphi(f, f')$. Hence, with $\epsilon' = \epsilon/\varphi_{\max}$, Algorithm 3 converges to an ϵ-optimal SSB strategy. □

4 PROOF OF CONCEPT

Although SSB encompasses many different criteria, we focus here on the probabilistic dominance criterion as it is an important case for which no model-free algorithm that is provably correct has been proposed. We plot here the results of four experiments: "sequential Gardner dice", "who wants to be a millionaire", "intransitive grid" and "race against the clock" using the probabilistic dominance criterion (hence values are all in $[-1, 1]$). For all runs, $10,000,000$ steps were performed in the MDP, ϵ was set to 0.1, and α_n was set to $1/(\eta_{s,a}+1)^{11/20}$.

Gardner Dice. We first present the results on Gardner's dice problem as formalized in Example 1. Figure 1(a) shows the evolution of the frequencies (f_A, f_B, f_C) with which each die has been played for a representative run. The optimal frequency vector $\mathbf{p}^* = (3/13, 3/13, 7/13)$ is shown as a green dot and the same vector biased by exploration $\mathbf{p}_\epsilon^* = (1-\epsilon) * \mathbf{p}^* + \epsilon * \mathbf{p}^{\pi_\mathcal{U}}$ by a red dot; the vector (f_A, f_B, f_C) tends towards $\mathbf{p}_\epsilon^*$, drawing triangles of decreasing surface around $\mathbf{p}_\epsilon^*$. Figure 1(b) presents the evolution of the Q-values of the three actions, a_A, a_B and a_C. One can see that the best die alternates between the three dice and that $\max\{Q_A, Q_B, Q_C\}$ tends towards 0. (The best response is always deterministic and so must be one of π_A, π_B, π_C. However at convergence its value has to be 0.)

Who wants to be a millionaire. In this popular television game show, a contestant answers 15 multiple-choice questions (with four possible answers) of increasing difficulty, for increasingly large sums, roughly doubling the pot each question. At each time step, the contestant may decide to walk away with the money currently won. If she answers incorrectly, then all winnings are lost except what has been earned at a "guarantee point" (questions 5 and 10). The player is allowed 3 lifelines (50:50, removing two of the choices, ask the audience and call a friend for suggestions); each can only be used once. We used the first model of the Spanish 2003 version of the game presented by Perea and Puerto [2007]. The probability of answering correctly is a function of the question's number and increased by the lifelines used (if any).

Intransitive grid. In this domain we study an episodic grid MDP containing 9 states. The agent always starts an episode in the bottom-right corner of the grid. Three terminal states, f_1, f_2 and f_3 can be attained at the three other corners of the grid. The agent can only go left and up. With a probability of 0.2, the agent makes a mistake and goes in the wrong direction. The preference relation between the final states is the following: $f_1 \succ f_2 \succ f_3 \succ f_1$.

Race against the clock. Lastly, we discuss the domain discussed in Example 3.[5] The racing circuit is represented by 6 physical positions $\{p_1, \ldots, p_6\}$ plus one, p_{el}, representing elimination. A state of the MDP is a triple (p, s, t) giving the current position $p \in \{p_1, \ldots, p_6, p_{el}\}$, the current speed of the car $s \in \{Slow, Medium, Fast\}$ and the current time $t \in \mathbf{N}$. In each state the agent can decide between 3 actions: accelerating, decelerating or keeping the same speed. At each time step t, the probability of running off the track is a function of s_t, p_t and a_t, taking into account both the speed of the car and the difficulty of the current part of the circuit. Finally, the time spent between two positions decreases stochastically with the current speed.

Results. Figure 1(c)-(f) shows the evolution of $L(\mathbf{p}_n^{real})$ and $L(\mathbf{p}_n)$ (i.e., the values of the optimal policies regarding reward functions defined by $\Phi\mathbf{p}_n^{real}$, $\Phi\mathbf{p}_n$) for each domain. The results are averaged over 20 runs. As expected the value of $L(\mathbf{p}_n)$ tends towards 0 as the number of learning steps increase. The value of $L(\mathbf{p}_n^{real})$ decreases and is much lower than ϵ ($= 0.1$ in the experiments) on all figures.

Finally, in Table 2, we compare the "race against the clock" results obtained by our SSB Q-learning algorithm to the results obtained by a standard Q-learning algorithm launched with three different reward functions $\mathbf{R}_1$, $\mathbf{R}_2$ and $\mathbf{R}_3$. For each reward function, a penalty of value $-t$ is received by the agent each time the circuit is completed in time t. For reward function $\mathbf{R}_i$, an elimination results in a penalty of value $r_{elem}^i \in \{-10, -25, -40\}$. For each algorithm we give the frequency of elimination f_{elem} and the average time T_{cc} of circuit completion. The final columns show the probabilities with which the SSB Q-learning agent would

[5] The complete description is given as supplementary material at hugogilbert.pythonanywhere.com.

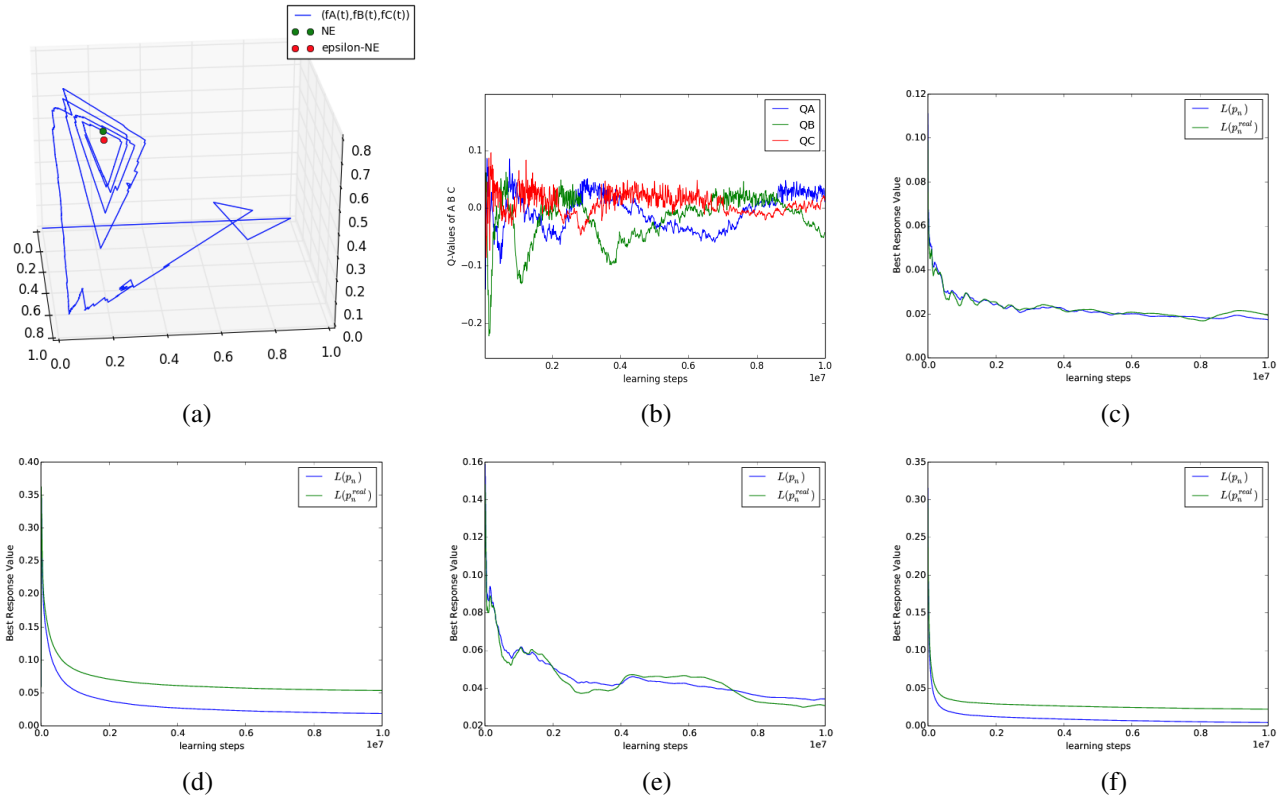

Figure 1: For the Gardner dice domain, **(a)** convergence of the policy in the space of die frequencies, **(b)** evolution of Q-values, **(c)** evolution of loss; For **(d)** Who wants to be a millionnaire, **(e)** Intransitive grid and **(f)** Race against the clock, evolution of loss.

beat ($\mathbf{P}_{\succ}$) and at least tie with ($\mathbf{P}_{\succeq}$) each Q-learning agent. As expected, f_{elem} decreases and T_{cc} increases with the penalty value of an elimination. For a Q-learning agent, this penalty would have to be tuned to give the best compromise. The SSB Q-learning agent does not face this problem and the last two columns show that this agent is more likely to produce a preferred episode.

5 Conclusion

Skew-Symmetric Bilinear (SSB) utility is a useful general decision model that encompasses many decision criteria (e.g., EU, threshold probability, probabilistic dominance, etc.). We designed a model-free reinforcement learning algorithm to compute an epsilon SSB-optimal policy and provided experimental results.

Table 2: Comparisons of SSB Q-Learning with Three Q-Learning Algorithms (Results Averaged Over 20 Runs).

	f_{elem}	T_{cc}	$\mathbf{P}_{\succ}$	$\mathbf{P}_{\succeq}$
SSB Q-learning	0.41	5.29	–	–
Q-learning with $\mathbf{R}_1$	0.48	4.34	0.37	0.64
Q-learning with $\mathbf{R}_2$	0.31	7.14	0.48	0.66
Q-learning with $\mathbf{R}_3$	0.26	9.09	0.54	0.66

Our current work can be extended in several natural ways. For instance, it would be interesting to tackle non-episodic problems. Another direction is to use more elaborate RL algorithms than Q-learning for computing best responses.

Our work is currently being applied in an industrial context with good results. An automated information extraction (IE) treatment chain is modelled as an MDP, and improved using a reward function balancing extraction quality and treatment time. SSB utility theory is used to formalise qualitative preferences expressed by human operators on the output of the treatments.

Acknowledgement

Work supported by the French National Research Agency through the Idex Sorbonne Universités, ELICIT project under grant ANR-11-IDEX-0004-02.

References

[Akrour *et al.*, 2012] R. Akrour, M. Schoenauer, and M. Sebag. APRIL: active preference-learning based reinforcement learning. *CoRR*, abs/1208.0984, 2012.

[Benaïm *et al.*, 2006] M. Benaïm, J. Hofbauer, and S. Sorin. Stochastic Approximations and Differential

Inclusions, Part II: Applications. *Mathematics of Operations Research*, 31(4):673–695, November 2006.

[Blavatskyy, 2006] P. Blavatskyy. Axiomatization of a Preference for Most Probable Winner. *Theory and Decision*, 60(1):17–33, 02 2006.

[Borkar, 1997] V. S. Borkar. Stochastic approximation with two time scales. *Systems & Control Letters*, 29(5):291 – 294, 1997.

[Borkar, 2008] V. S. Borkar. *Stochastic approximation : a dynamical systems viewpoint*. Cambridge university press New Delhi, Cambridge, 2008.

[Brown, 1951] G. W. Brown. Iterative solution of games by fictitious play. *Activity Analysis of Production and Allocation*, 1951.

[Busa-fekete *et al.*, 2014] R. Busa-fekete, B. Szörényi, P. Weng, W. Cheng, and E. Hüllermeier. Preference-based evolutionary direct policy search, 2014.

[Chvátal, 1983] V. Chvátal. Matrix games. In *Linear programming*, chapter 15, pages 228–239. Freeman, 1983.

[Dudík *et al.*, 2015] M. Dudík, K. Hofmann, R. E. Schapire, A. Slivkins, and M. Zoghi. Contextual dueling bandits. *CoRR*, abs/1502.06362, 2015.

[Fishburn, 1984] P. Fishburn. SSB utility theory: an economic perspective. *Mathematical Social Sciences*, 8(1):63 – 94, 1984.

[Fishburn, 1991] P. Fishburn. Nontransitive preferences in decision theory. *Journal of Risk and Uncertainty*, 4(2):113–134, 1991.

[Furnkranz *et al.*, 2012] J. Furnkranz, E. Hullermeier, W. Cheng, and S. Park. Preference-based reinforcement learning: a formal framework and a policy iteration algorithm. *Machine Learning*, 89(1-2):123–156, 2012.

[Gilbert *et al.*, 2015a] H. Gilbert, O. Spanjaard, P. Viappiani, and P. Weng. Solving MDPs with Skew Symmetric Bilinear Utility Functions. In *IJCAI*, 2015.

[Gilbert *et al.*, 2015b] H. Gilbert, O. Spanjaard, P. Viappiani, and P. Weng. Reducing the number of queries in interactive value iteration. In *ADT*, pages 139–152, 2015.

[Hofbauer, 1995] J. Hofbauer. Stability for the best response dynamics. *Working Paper*, 1995.

[Kalathil *et al.*, 2014] D. M. Kalathil, V. S. Borkar, and R. Jain. A learning scheme for approachability in MDPs and Stackelberg stochastic games. *CoRR*, abs/1411.0728, 2014.

[Mas-Colell *et al.*, 1995] A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic theory*. Oxford University press, New York, 1995.

[Nakamura, 1989] Y. Nakamura. Risk attitudes for nonlinear measurable utility. *Annals of Operations Research*, 19:pp. 311–333, 1989.

[Perea and Puerto, 2007] F. Perea and J. Puerto. Dynamic programming analysis of the TV game who wants to be a millionaire? *European Journal of Operational Research*, 183(2):805 – 811, 2007.

[Puterman, 1994] M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994.

[Rivest and Shen, 2010] R. Rivest and E. Shen. An optimal single-winner preferential voting system based on game theory. In V. Conitzer and J. Rothe, editors, *Proceedings Third International Workshop on Computational Social Choice*. Düsseldorf University Press, 2010.

[von Neumann and Morgenstern, 1947] J. von Neumann and O. Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1947.

[Watkins and Dayan, 1992] C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.

[Weng and Zanuttini., 2013] P. Weng and B. Zanuttini. Interactive value iteration for Markov decision processes with unknown rewards. In *International Joint Conference in Artificial Intelligence*, 2013.

[Weng *et al.*, 2013] P. Weng, R. Busa-Fekete, and E. Hüllermeier. Interactive Q-Learning with Ordinal Rewards and Unreliable Tutor. In *ECML/PKDD Workshop Reinforcement Learning with Generalized Feedback*, 2013.

[Wilson *et al.*, 2012] A. Wilson, A. Fern, and P. Tadepalli. A Bayesian approach for policy learning from trajectory preference queries. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1133–1141. Curran Associates, Inc., 2012.

[Wirth and Fürnkranz, 2013] C. Wirth and J. Fürnkranz. EPMC: every visit preference Monte Carlo for reinforcement learning. In *Asian Conference on Machine Learning, ACML 2013, Canberra, ACT, Australia, November 13-15, 2013*, pages 483–497, 2013.

[Wirth and Neumann, 2015] C. Wirth and G. Neumann. Model-free preference-based reinforcement learning. In *EWRL*, 2015.

[Yu *et al.*, 1998] S. Yu, Y. Lin, and P. Yan. Optimization models for the first arrival target distribution function in discrete time. *Journal of Mathematical Analysis and Applications*, 225(1):193–223, 1998.